Experiment 7: Solving Equations of Linear Algebra

Objective of the Experiment

After completing this experiment, students will be able to:

a) To implement the Gauss Elimination method using MATLAB programming.
b) To apply forward elimination and backward substitution in MATLAB to solve systems of linear equations.
c) To develop proficiency in MATLAB matrix operations and loop-based numerical computation.

Introduction

The Gauss Elimination method is a fundamental numerical technique used to solve systems of simultaneous linear equations. The method systematically reduces the given system into an equivalent form that is easier to solve.

The procedure involves two main stages:

Stage-1: Forward Elimination, where the augmented matrix is transformed into an upper triangular form.

Stage-2: Backward Substitution, where the unknown variables are calculated starting from the last equation.

Working Rule

Consider the system of equations

$$a_{11}x + a_{12}y + a_{13}z = b_1$$
$$a_{21}x + a_{22}y + a_{23}z = b_2$$
$$a_{31}x + a_{32}y + a_{33}z = b_3$$

In matrix form

$$AX = B$$
$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

Augmented matrix

$$C = [A:B] = \begin{bmatrix} a_{11} & a_{12} & a_{13}: b_1 \\ a_{21} & a_{22} & a_{23}: b_2 \\ a_{31} & a_{32} & a_{33}: b_3 \end{bmatrix}$$

Reduce the augmented matrix to echelon form using elementary row transformations,

$$C = \begin{bmatrix} c_{11} & c_{12} & c_{13}: d_1 \\ 0 & c_{22} & c_{23}: d_2 \\ 0 & 0 & c_{33}: d_3 \end{bmatrix}$$

After Gauss elimination, the system becomes upper triangular:

$$c_{11}x + c_{12}y + c_{13}z = d_1$$
$$c_{22}y + c_{23}z = d_2$$
$$c_{33}z = d_3$$

Start from the last equation (only one unknown): $z = \frac{d_3}{c_{33}}$

Substitute $z$into the second equation: $y = \frac{d_2 - c_{23}z}{c_{22}}$

Substitute $y$and $z$into the first equation:$x = \frac{d_1 - c_{12}y - c_{13}z}{c_{11}}$

MATLAB Implementation

Problem: Gauss Elimination to Find an Augmented Upper Triangular Matrix.
Given the system of linear equations:

$$\begin{cases} 6x_1 + 1x_2 - 6x_3 - 5x_4 = 6 \\ 4x_1 - 3x_2 + 0x_3 + 1x_4 = -7 \\ 2x_1 + 2x_2 + 3x_3 + 2x_4 = -2 \\ 0x_1 + 2x_2 + 0x_3 + 1x_4 = 0 \end{cases}$$

Task: Use the Gauss Elimination method to convert the augmented matrix of the system into upper triangular form.

Coefficient matrix A:

$$A = \begin{bmatrix} 6 & 1 & -6 & -5 \\ 4 & -3 & 0 & 1 \\ 2 & 2 & 3 & 2 \\ 0 & 2 & 0 & 1 \end{bmatrix}$$

RHS vector C:

$$C = \begin{bmatrix} 6 \\ -7 \\ -2 \\ 0 \end{bmatrix}$$

Augmented matrix [A|C] combines them into **one matrix**:

$$\text{Aug} = \begin{bmatrix} 6 & 1 & -6 & -5 & 6 \\ 4 & -3 & 0 & 1 & -7 \\ 2 & 2 & 3 & 2 & -2 \\ 0 & 2 & 0 & 1 & 0 \end{bmatrix}$$

```
% GAUSS ELIMINATION METHOD

% This program solves a system of linear equations
% using Forward Elimination and Backward Substitution

clc;        % Clear the command window
clear;      % Remove all variables from memory

% STEP 1: Define the coefficient matrix (A) and RHS vector (C)

% A contains the coefficients of the variables
% C contains constant terms of equations

A = [6 1 -6 -5; 4 -3 0 1; 2 2 3 2; 0 2 0 1];   % Coefficient matrix

C = [6; -7; -2; 0];                       % Right-hand side vector

% STEP 2: Check if the matrix is square

% Gauss Elimination works only for square matrices

[m, n] = size(A);     % Get number of rows (m) and columns (n)
if m ~= n
   dis('Matrix must be square to apply Gauss Elimination');
end

% STEP 3: Form the augmented matrix [A | C]

% This combines coefficients and constants into one matrix

Aug = [A C];
disp('Initial Augmented Matrix [A | C]:');
disp(Aug);
```

```
% STEP 4: FORWARD ELIMINATION

% Purpose:
% Convert matrix into upper triangular form
% (All elements below the main diagonal become zero)

for k = 1: n-1          % k selects the pivot position (diagonal element)
                        % Pivot element = Aug(k,k)
   for i = k+1: n       % i selects rows below the pivot row
                        % These rows will be modified to make zeros
      factor = Aug(i,k) / Aug(k,k);     % Compute the elimination factor
                                        % This factor tells how much of the pivot row to subtract
      for j = k: n+1    % j selects columns to be updated
                        % Start from pivot column and include RHS (n+1)
         Aug(i,j) = Aug(i,j) - factor * Aug(k,j);  % Perform row operation:
                                                   % Row_i = Row_i - factor × Row_k
      end
   end
end
disp('Upper Triangular Augmented Matrix:');
disp(Aug);
```

Explanation

for k = 1: n-1
k represents the pivot step
k = 1 → eliminate terms below a11
k = 2 → eliminate terms below a22
…
k = n−1 → eliminate terms below a (n−1, n−1)
No work is needed for k = n, because nothing is below the last row

for i = k+1: n
i represents the row being updated in the current pivot step
i = k+1 → first row below the pivot
i = k+2, k+3 … n → remaining rows below the pivot
Pivot row k and rows above are not changed

for j = k: n+1

j represents the column being updated in the current row i
j = k → pivot column, to eliminate the element
j = k+1, k+2 … n+1 → remaining matrix entries that must be updated

```
% STEP 5: BACKWARD SUBSTITUTION

% Purpose:
% Solve variables starting from the last equation

x = zeros(n,1);              % Initialize solution vector

x(n) = Aug(n, n+1) / Aug(n,n);          % Solve the last variable directly

                                         % Last equation has only one unknown

% Solve remaining variables one by one (moving upward)

for i = n-1: -1: 1
    sum = 0;                  % Reset summation for each equation
    for j = i+1: n            % Add known variable contributions
        sum = sum + Aug(i,j) * x(j);
    end
    x(i) = (Aug(i,n+1) - sum) / Aug(i,i);   % Compute the current variable
end

% STEP 6: Display the solution

disp('Solution of the system [x1, x2, ..., xn]:');
disp(x);
```

Explanation

for i = n-1: -1: 1
selects the current row (variable) to solve.

- i = n-1 → solve second-to-last variable x(n-1)
- i = n-2 → solve third-to-last variable x(n-2)
- …
- i = 1 → solve first variable x(1)

for j = i+1: n
sums contributions from already known variables in row i.

- j = i+1 → first known variable to the right of pivot in row i
- j = i+2, i+3 … n → remaining known variables in the row

Practice Problem

Problem 1

Solve the following system of linear equations using the Gauss Elimination method:

$$\begin{aligned} 2x_1 + 2x_2 + x_3 &= 6 \\ 4x_1 + 2x_2 + 3x_3 &= 4 \\ x_1 - x_2 + x_3 &= 0 \end{aligned}$$

Problem 2

Solve the following system of linear equations using the Gauss Elimination method:

$$\begin{aligned} x_1 + x_2 + x_3 + x_4 &= 10 \\ 2x_1 + 3x_2 + x_3 + 2x_4 &= 20 \\ x_1 + 2x_2 + 3x_3 + x_4 &= 16 \\ 3x_1 + x_2 + 2x_3 + 2x_4 &= 19 \end{aligned}$$

Problem 3

Solve the following system of linear equations using the Gauss Elimination method:

$$\begin{aligned} 4x_1 - x_2 + 2x_3 &= 1 \\ 3x_1 + 2x_2 - x_3 + x_4 &= 4 \\ 2x_1 - 3x_2 + x_3 + 2x_4 &= -2 \\ x_1 + x_2 + x_3 + x_4 &= 6 \end{aligned}$$